

Experiencias de programación en las escuelas

María Cecilia Martínez¹
María Emilia Echeveste

Resumen

En los últimos años, diferentes organismos han promovido la enseñanza de la programación en la escuela. En esta publicación, analizaremos algunas experiencias de enseñanza de la programación en escuelas públicas primarias y secundarias de la provincia de Córdoba que se desplegaron como resultado de cursos cortos de formación docente. Desde un enfoque exploratorio identificamos, a partir de reflexiones docentes y observaciones de clase, cuatro emergentes en torno a experiencias de enseñanza de la programación con videojuegos: 1) el entusiasmo que genera en los estudiantes, 2) la posibilidad de integrar disciplinas a partir de proyectos de programación, 3) el desarrollo del trabajo colaborativo y 4) la inclusión en tareas de programación de estudiantes con diferentes capacidades cognitivas. Se ofrece una sistematización de las reflexiones de los docentes y algunos registros de observación que permitieron construir estos emergentes.

Didáctica de la computación – integración – enseñanza de la programación

Programming teaching experiences in schools

Abstract

During the last decade different organization have promoted Computer Science education in mandatory schooling. In this paper we analyze some teaching experiences in programming in primary and secondary public schools from Córdoba, Argentina. These experiences resulted form short professional

¹ Universidad Nacional de Córdoba. CE: cecimart@gmail.com

development courses. Following an exploratory approach, we relied on teacher's reflections and classroom observation to identify for emerging themes around teaching programming through video games: 1) student enthusiasm, 2) the possibility of integrating different disciplines in programming projects, 3) collaborative work among the students, 4) the inclusion of special education students in programming tasks. We systematized teacher's reflections and classroom observation that allowed to built these emerging themes.

Computer science education – integration – teaching of programming

Introducción

Actualmente, la tecnología se incorpora en las escuelas de manera creciente, no solo en sus modos de uso (que cada vez son de mayor calidad), sino también en la apropiación de sus contenidos como objeto de estudio. Esta publicación analizará algunos modos de enseñar programación y su potencialidad para articular esta área del conocimiento con diferentes asignaturas. Mostraremos que –bajo ciertas condiciones– la enseñanza de la programación convoca a los estudiantes a una relación con el conocimiento que se caracteriza por el interés en la aprender para construir, la integración interdisciplinar, la colaboración y la inclusión de niños y niñas con problemas de aprendizaje. Todas estas características tensionan el formato escolar tradicional. Las experiencias que analizamos, han permitido observar movimientos en los alumnos en relación a sus modos de abordar el trabajo en aula a partir de un proyecto de programación de videojuegos.

El contexto de la enseñanza de la programación

Es constante y creciente la apropiación de la programación y la tecnología en las escuelas. Diversos estudios muestran que año tras años las y los docentes incorporan la computadora de manera cada vez más significativa en sus clases (Casablanca, Caldeiro y Odetti, 2016; Lago Martínez, 2015). Como resultado del programa “Conectar Igualdad”, Casablanca (2016) y sus colegas encontraron que los docentes usan la computadora para ofrecer propuestas de enseñanza donde se incluyen a las Tecnologías de la Información y la Comunicación (TIC). Asimismo, cuando la propuesta lo requiere, se realizan usos de la tecnología que implican conocer programas sofisticados para desarrollar contenidos interactivos. Este incremento en la apropiación de la tecnología se da tanto en el aumento de frecuencia de uso como en la calidad de las aplicaciones que se trabajan en la escuela. Este salto cualitativo en el aumento de la informática en las escuelas nos sitúa en un momento diferente para recibir la incorporación de los programas de distribución de equipamiento y formación docente masivos en TIC.

Desde el 2005, han cobrado vigencia en la enseñanza de la computación los enfoques “lingüísticos” (Levis, 2007) que proponen enseñar lenguajes y conceptos de programación para entender no sólo cómo funciona una computadora, sino crear nueva tecnología a partir de desarrollar programas de computación. Los enfoques lingüísticos centran su mirada en la promoción de construcción de saberes, resolución de problemas y desarrollo del pensamiento analítico y crítico. Diferentes sectores vinculados con la educación se han unido para fomentar que la computadora sea “objeto de estudio”, y las ciencias de la computación la disciplina encargada de abordar ese saber (por ejemplo, la iniciativa Program.ar, Code.org, Programamos, etcétera).

A partir de informes que emitieron varios sectores preocupados por la enseñanza de la programación en las escuelas, muchos países han avanzado en acordar y definir qué conceptos de computación serían centrales para enseñar en las escuelas y en qué secuencia. En nuestro país, hay documentos emitidos por el Iniciativa Program.AR de la Fundación Sadosky y el Consejo Universitario de Ciencias Exactas y Naturales (CUCEN) que orientan la selección de contenidos. Para 2019, el Consejo Federal de Educación aprobó los Núcleos de Aprendizaje Prioritarios (NAPs) en Robótica y Programación. También para 2019, la Fundación Sadosky junto con otras universidades nacionales publicaron dos manuales para enseñar a programar en el nivel primario. Todos estos esfuerzos van orientando los contenidos para incluir la programación en las escuelas.

Asimismo, durante los últimos cinco años, grupos en diferentes universidades del mundo han desarrollado varias plataformas didácticas para enseñar a programar. Estas plataformas, por lo general cuentan con una estética atractiva y lúdica que sumado a su *feedback* inmediato favorecen la interacción de los estudiantes con las animaciones y los ejercicios.

De esta manera, consideramos entonces que hay tres situaciones clave que nos permiten pensar en la posibilidad de enseñar programación en el contexto de la actual escuela: 1) la creciente apropiación que hacen docentes y estudiantes de la computadora y las tecnologías en general en el ámbito escolar, 2) los desarrollos curriculares, 3) los avances en el desarrollo de las plataformas didácticas para enseñar a programar. Veamos de qué hablamos, cuando hablamos de programación.

Definiciones conceptuales sobre la programación

Programar es definido generalmente como la acción de darle instrucciones precisas a una máquina en un lenguaje que la computadora pueda entender para que ejecute y automatice alguna acción o conjunto de acciones. Automatizar es hacer que una máquina repita tareas de manera más rápida y eficiente de lo que lo harían los humanos (Lee et al, 2011). Para Papert (1980), además, programar es esencialmente crear con tecnología y construir con una computadora. Y más allá de ofrecer instrucciones precisas, la construcción es representación de ideas a través de un lenguaje: es esencialmente expresar una forma de

resolver un problema a partir de la creación de un artefacto. Pensemos por ejemplo en la construcción de una habitación. En ella está involucrada el propósito de la construcción de esa habitación, para qué se usará, la técnica constructiva, los detalles de terminación que permitirán explotar las funciones de la habitación, los materiales, la estética, la robustez, la posibilidad de replicarla, etcétera. Del mismo modo, programar es construir.

Por ello, cuando se habla de programación se describe a una disciplina que requiere integrar simultáneamente el uso de cierto grado de creatividad, un conjunto de conocimientos técnicos específicos de computación y la capacidad de operar constantemente con abstracciones (Martínez López, Bonelli y Sawady, 2014). Los programadores se dedican principalmente a construir programas, es decir, un texto descriptivo que al ser procesado por una computadora da solución a un problema propuesto por los humanos. Vemos entonces qué inherente a la programación está la apropiación de un lenguaje formal y su uso para transmitir ideas. Se ha discutido bastante en el campo de la educación la importancia del lenguaje para desarrollar el pensamiento desde el trabajo de Lev Vigotsky (1999) en adelante. Sabemos que el lenguaje contribuye, en primer lugar, en abstraer objetos de símbolos. El lenguaje también requiere de ordenar y jerarquizar ideas, desarrollar el pensamiento simbólico, crear y comunicar. Los lenguajes formales, como los que se usan en programación, requieren además abstraerse del significado del lenguaje para que los símbolos sean operados como objetos matemáticos. Este proceso de abstraerse de la semántica en el uso de un lenguaje ha sido denominado “des-semantización” (Dutilh Novaes, 2011). Para Novaes, el razonamiento con lenguajes formales permite revisar creencias y argumentos, es decir, razonar sobre nuestras posiciones debido al importante componente de lógica que tienen. La enseñanza de la programación contribuye entonces a desarrollar este tipo de pensamientos.

Respecto de los conceptos propiamente informáticos, Aloi (2016) considera que

La educación en programación requiere del dominio de diferentes herramientas informáticas tales como lenguajes de programación, bibliotecas, entornos de desarrollo, compiladores, etcétera, que se suman a lo más importante que es la comprensión y utilización de herramientas conceptuales, de conocimientos y de estrategias que conforman una manera de hacer y sobre todo una manera de pensar (p. 210).

Es decir, la programación integra el uso de un lenguaje para aplicar conceptos de lógica en la resolución de situaciones que requieren procesar información. Es justamente la puesta en juego de conceptos lógicos y abstracción a través de una máquina para obtener un resultado lo que diferencia al pensamiento computacional de las TIC, cuyo énfasis está en el uso de tecnología previamente programada (Grover y Pea, 2013).

Advertimos que estas definiciones sobre la programación se ligan íntimamente con los objetivos de la educación general de Nivel Primario y Secundario. Resolver problemas, representar información a través de diferentes lenguajes, producir enunciados y diferentes tipos de textos, encontrar regularidades, descomponer un problema, etcétera, son

aprendizajes básicos que se espera desarrollar tanto en la educación primaria y secundaria y donde el aprendizaje de la programación puede hacer un aporte.

Zapata Ros (2015) nos advierte de que no basta incluir programación en el currículum para desarrollar el pensamiento computacional, sino que es necesario también abordar su enseñanza de manera que permita la construcción, la creación, la expresión, la representación, la resolución de problemas, la abstracción, y la descomposición de un problema mayor en sub problemas. Recuperando la conceptualización de la programación en tanto integración de habilidades y saberes para construir resoluciones a problemas computacionales, es que nos interesa indagar cómo los docentes integran las experiencias de enseñanza de la programación a sus clases.

Metodología

Este trabajo exploratorio busca documentar algunas prácticas de enseñanza de la programación en contextos de escuelas primarias y secundarias públicas de la mano de docentes que no son especialistas en computación y que recibieron un curso de capacitación en la universidad o el gremio docente (UEPC), de 50 horas en enseñanza de la programación a través del desarrollo de videojuegos. Este estudio se centra en los datos recabados de una centena de docentes de cuatro regiones de la provincia de Córdoba, que participaron del curso de enseñanza de la programación ofrecido por UEPC y por la Universidad Nacional de Córdoba (UNC). Dos de estas regiones se encuentran en contextos rurales y las otras dos en contextos urbanos.

En el curso de capacitación se presentaron distintas plataformas educativas de software libre, dos de ellas creadas por la UNC. Trabajamos con Introducción a la robótica con la plataforma UNC++Duino, nociones de autómatas de chat con la Plataforma Chatbot y la creación y programación de animaciones y videojuegos con las plataformas Alice y Scratch.

Estas plataformas, por lo general, recuperan un aspecto lúdico, son ricas visualmente y permiten a los estudiantes realizar ejercicios o animaciones y ver sus resultados rápidamente. El diseño de estas plataformas se vio enriquecido con el concepto de intuición para aprender computación (Guzdial, 2008). Estos estudios mostraron que la mayoría de adultos y niños cuando se ponían en situación de resolver un problema lógico matemático (tales como los problemas que resuelve la informática), usaban un lenguaje natural, es decir el lenguaje coloquial, que incluía algunas nociones previas de lógica. Por ejemplo, la noción de condicional se puede usar para indicar una condición entre una proposición y una acción tal como “si se da la condición A, entonces se hace B”. Este tipo de estudios ha contribuido al desarrollo de plataformas que invitan a “aprender haciendo” desde las intuiciones o saberes previos. Estas plataformas tienen el propósito de ser una herramienta para desarrollar un proyecto específico tal como programar un videojuego, una animación, un robot que responde chats de manera automática, el recorrido de un robot en un laberinto, etcétera. Las funciones que tiene la plataforma se ofrecen en “bloques de arrastre” que ya tienen previamente programada

la función que realiza el programa. Los estudiantes van realizando combinaciones de bloques que al ejecutarse muestran una secuencia de líneas de código de un programa. Los alumnos y alumnas ejecutan inmediatamente el programa observando errores y aciertos.

Para esta ocasión, se analizaron las notas de clases que los docentes formadores fuimos registrando en cada encuentro, ya que los docentes que realizaron el curso reflexionaban constantemente sobre su práctica, y completaron en sus evaluaciones finales un apartado en el que abordaba el análisis de su experiencia desde las dimensiones de conocimientos previos, el clima de la clase, expectativas y rendimiento en general y saberes de otras materias que se integraron. Esta información se complementó con los registros de tutores, conformados por estudiantes avanzados de la carrera de Ciencias de la Computación (Facultad de Matemática, Astronomía y Física –FaMAF), quienes acompañaban las prácticas de los docentes en sus escuelas.

Resultados

Sobre los docentes y sus motivaciones para introducir la programación en el aula

Los docentes que participaron de nuestros cursos pertenecían tanto a instituciones primarias como secundarias, y enseñaban diversas materias. La gran mayoría de los docentes asistentes no sabían de qué se trataba la programación. Un tema emergente durante el primer encuentro fue que muchos de los cursantes estaban sorprendidos cuando hacíamos la distinción entre TIC y programación. En general, los docentes comprendieron la importancia de integrar las tecnologías y saberes básicos de programación a la educación, y de la mano del curso fueron descubriendo herramientas y modos posible de hacerlo, incluso aquellos docentes de materias no tan afines a la computación como educación física o inglés, que en un inicio se sentían preocupados por llevar estos contenidos al aula.

Cuando indagábamos cuáles eran las motivaciones de asistir al curso, muchos docentes manifestaron la necesidad de capacitarse en tecnologías. Especialmente en regiones alejadas de las grandes ciudades donde no habían tenido ofertas de cursos con TIC. Sin embargo, el principal estímulo para aprender tecnología, según los relatos de los docentes, era la necesidad de vincularse con sus estudiantes a través de artefactos que son atractivos para los jóvenes. Los docentes manifestaban que a sus alumnos les interesa mucho trabajar con computadoras y teléfonos celulares –que también son computadoras– y ofrecer actividades que se puedan resolver con estos dispositivos podría contribuir a mejorar la motivación y la comunicación entre profesores y estudiantes.

Los docentes identificaron, en sus reflexiones escritas al final de curso, que la mayoría de sus estudiantes tenía conocimientos sobre TIC (uso de programas de oficina, de redes sociales y de juegos por computadora), pero no tenían saberes previos de programación. Este análisis permitió trabajar algunas creencias que traían los docentes sobre la imposibilidad de conectarse o enseñar tecnologías por sostener que las y los alumnos saben más que ellos.

En efecto, en las primeras experiencias de enseñanza de la programación por docentes

de grado, Papert (1995) relata que la conciencia de ser profesores les impedía entregarse por completo a la experiencia de aprendizaje de la programación orientada por una construcción colaborativa, donde los estudiantes pueden recibir pero también pueden dar. Beatriz Greco (2012) también tensiona la idea de pensar disociadamente que quien enseña no es pensando aprendiendo y viceversa.

Sobre la puesta en práctica de clases de programación

El curso de capacitación requería poner en práctica al menos una clase de programación. La mayoría de los docentes (el 70%) cumplió con este requisito. Quienes no pudieron, ya sea por falta de equipamiento, porque no tenían un curso a cargo, por falta de tiempo o imposibilidad de sumar o integrar contenidos a su materia, presentaron un plan de trabajo.

Los docentes que pusieron en práctica proyectos de programación reportaron que entre un 85% a 90% de los alumnos de su clase pudo terminar la tarea planteada, es decir, terminaron sus videojuegos. En el caso de un segundo grado donde solo había 3 computadoras, solo el 50% de los estudiantes terminaron el juego. Nos preguntamos si fue la escasez de equipamiento o la baja edad del alumnado lo que limitó la cantidad de estudiantes que terminaron el videojuego.

Para un 20% de los docentes, los estudiantes superaron las expectativas de su clase. Estos grupos lograron avanzar más allá de la consigna descubriendo nuevas funciones en la plataforma para hacer un videojuego más complejo al propuesto inicialmente. Para el resto, los estudiantes cumplieron con las expectativas logrando terminar el trabajo, pero sin enriquecerlo con funciones extras.

En relación a la clase de programación, todos los docentes describieron el clima de la clase en términos positivos. “Entusiasmo” fue la palabra que más frecuentemente usaron para comentar cómo fue recibida la propuesta en aula (usada entre un 20% de los docentes). También se usaron frecuentemente las palabras de: curiosidad, alegría, asombro, sorpresa, investigación, exploración, se esforzaban, interés y desorden. Por ejemplo, una docente expresó:

El clima de la clase al comienzo fue bastante complicado, ellos no entendían el motivo de la utilización del programa porque en Ciudadanía y Participación estamos acostumbrados a trabajar de otra manera. Me costó hacerles entender el objetivo de la clase. Una vez que lo entendieron se motivaron y ahí sí, el clima cambió y se hizo una clase amena, pero sobre todo de interacción mutua entre ellos y yo. Y el rendimiento fue bueno, la mayoría de los alumnos se engancharon... todos terminaron el proyecto... la calidad fue buena, si bien hubo estudiantes que solo lo hicieron para cumplir, sé que la mayoría lo hizo a conciencia y se divirtió.

Otra docente relató:

Los alumnos se mostraron expectantes y ansiosos ante la posibilidad de poder crear sus propios videojuegos. La clase se realizó con mucha dinámica producida por sus propios

intereses. Los alumnos querían seguir estas producciones de videojuegos en otras asignaturas.

Los docentes manifestaron en general que al principio la clase les presentaba incertidumbre –que ellos describieron como desorden o caos– por presentar un desafío nuevo a los estudiantes tal como lo era desarrollar su propio videojuego. La propuesta a los docentes consistía en que, una vez presentada la plataforma, sus estudiantes comiencen a trabajar libremente. En investigaciones anteriores notamos que, por lo general, los estudiantes no desean ser interrumpidos por explicaciones de los docentes, sino que prefieren explorar la plataforma a medida que van construyendo su videojuego. Aprendizaje de la plataforma y construcción de videojuego van constituyendo un solo momento de la clase. En síntesis, la mayoría de los docentes pudo superar el desafío de poner en práctica una propuesta novedosa y realizar al menos una clase de programación, expresando también, que el clima de la clase fue de entusiasmo y de exploración activa.

Dinámicas que se generaron en los cursos

La introducción de experiencias acotadas de enseñanza de la programación en la escuela mostró que promueve el interés de los estudiantes, el trabajo en grupo, el buen desempeño de estudiantes que regularmente tienen dificultades y la integración de distintas disciplinas.

a. Interés por los estudiantes e integración entre disciplinas

En las experiencias que propusieron los docentes a sus alumnos se ofrecía un desafío para que los estudiantes elaboraran un proyecto de videojuegos o animaciones. Más que una consigna o una tarea, la propuesta era una invitación a crear un producto tecnológico significativo para ellos.

En el relato de uno de los tutores:

Comienza la clase indagando entre sus alumnos de 2do año sus saberes sobre videojuegos. Seguramente muchos de los estudiantes se habrán preguntado qué tendrá que ver la Química (materia a su cargo) con los videojuegos. Luego de recuperar algunas de sus experiencias como expertos “gamers”, Analía,² la profesora, les presenta una plataforma para programar videojuegos llamada Scratch y les muestra cómo cargarla en sus netbooks.

Les propone desarrollar un videojuego que complemente las actividades que están haciendo en la huerta de la escuela donde los estudiantes realizan un “Bio-insecticida” para mostrar en la Feria de Ciencias de la Escuela. El desafío propuesto era desarrollar un videojuego que permita eliminar insectos que afectan a la huerta, pero dejando vivos aquellos que no destruyen las plantaciones.

El aprendizaje por proyectos es una idea bastante elaborada de la didáctica general. Los

² Para preservar el anonimato de los sujetos se han cambiado los nombres por otros ficticios.

proyectos permiten crear artefactos, programas, bienes intangibles como textos o películas o realizar intervenciones sociales en donde los alumnos ponen a disposición e integran saberes de diferentes disciplinas que permiten elaborar ese proyecto. En la didáctica específica de la programación se habla de que los proyectos diseñados para enseñar a programar deberían tener **“piso bajo y techo alto”** (Resnick y Silverman, 2005). Para explicar esta intencionalidad didáctica Resnick utiliza la experiencia de construcción con los bloques de Rastri o Lego. Los Rastri consisten en una cantidad reducida de bloques de diferentes formas y tamaños. Con ellos se puede construir desde un puente con tres piezas, hasta ciudades en escala que son verdaderas obras de ingeniería. Los proyectos diseñados con el criterio “piso bajo-techo alto” son los que se pueden completar con programaciones sencillas que requieren pocas líneas de código para obtener rápidamente resultados –piso bajo– y al mismo tiempo ofrecen posibilidades de construir proyectos cada vez más complejos y sofisticados a medida que se explora el entorno y se conocen y combinan sus herramientas –techo alto. Además, permiten la alternancia en una amplia gama de exploraciones y habilitan una gran variedad de proyectos diferentes –paredes anchas. En este sentido, son accesibles y atractivas para niños con diferentes estilos de aprendizaje y modos de conocimiento (Resnick, 2008).

El trabajo por proyectos de programación permite integrar diferentes disciplinas –paredes anchas, y favorece que cada grupo de alumnos pueda avanzar según sus motivaciones –techo alto. En la programación, muchas veces es la misma práctica la que marca los errores y los aciertos a la hora de hacer funcionar un programa sin necesidad de una supervisión constante, en este caso del docente. El *feedback* inmediato que nos posibilita la computadora favorece no solo la exploración y descubrimiento de los contenidos, sino que les permite avanzar y otorgar autonomía y confianza a sus alumnos en la aprehensión de los conocimientos (Echeveste, 2017).

b. Trabajo en colaboración

En relación a las experiencias recuperadas, observamos que los docentes promueven instancias de intercambio y colaboración entre los estudiantes para que estos comenten los avances y dificultades de sus proyectos, así como también sus “descubrimientos”. Estos intercambios permiten que los estudiantes asistan a sus compañeros pensando estrategias para sortear problemas o ideas para continuar desarrollando cada proyecto.

Por ejemplo, en su reflexión sobre la clase, la profesora Analía comenta que el trabajo se realizó de manera grupal y en un clima de colaboración donde los estudiantes se ayudaban mutuamente. Tres grupos que continuaron trabajando fuera del espacio del aula, presentaron sus videojuegos terminados en la Feria Anual de la escuela.

Por otro lado, si bien la secuencia didáctica debe ser pensada, proyectada y planificada, el trabajo con proyectos propicia la aparición de situaciones no previstas. Como nos explica Papert (1995), cuando el maestro y el estudiante se enfrentan a un problema real que surge en el desarrollo de un proyecto, el problema es un reto para los dos y ambos pueden aportar lo suyo. Así, el docente ya no ocupa un lugar central como dador de conocimientos, sino que

se convierte en un coaprendiz. Este tipo de propuestas alienta el trabajo colaborativo donde, estudiantes entre sí y con el docente, comparten los genuinos descubrimientos que emergen como parte del desarrollo del proyecto.

c. Integración de estudiantes

Los relatos recuperados de las aulas sugieren que es posible repensar algunas concepciones de sujeto “educable” centrada en presunciones obtenidas de conductas que difieren a las esperadas por el sistema escolar tradicional. Los jóvenes catalogados previamente por el sistema o por sus docentes como “los que nunca se enganchan”, “los repitentes”, “los niños con retraso” han logrado un buen desempeño en las clases de programación, que sorprendió a sus docentes y superó el desempeño del resto de sus compañeros.

Veamos algunos fragmentos recuperados de los registros de los tutores:

Nacho, un nene “integrado”, supuestamente tiene cierto nivel de retardo, así me lo plantearon, mientras todos hacían las actividades estaba solo sentado al lado de su grupo (...) entonces le ofrecí acompañarlo a hacerla. Nos sentamos en un costado los dos y la verdad que el nene “la hizo de goma” literalmente, lo hizo todo solo y súper rápido. Es más, en la página 8 del libro, al final hay un ejercicio extra para “expertos”, y como había terminado antes que los otros, le pregunté “¿te animas a hacerlo?”, me dijo “sí” (con un poquito más de confianza) y como no le salía muy bien así en el aire el ejercicio, ¡lo representó con lápices de distintos tamaños! Sinceramente ¡quedé sorprendido! Yo sin conocimientos psicológicos, me atrevería a decir que ese chico no tiene retraso, pero eso queda en manos de los especialistas.

Relatos como estos son frecuentes en la mayoría de las clases donde los docentes capacitados ofrecen experiencias de programación. Nuestra hipótesis es que la programación requiere desplegar funciones cognitivas diferentes a las que el sistema tradicional demanda, ofreciendo la posibilidad de expresarse, a través de la construcción de un proyecto, a estos estudiantes que tienen mejor desarrollo en esas funciones.

Reflexiones finales

En este escrito presentamos algunos temas emergentes a partir de analizar experiencias de enseñanza de la programación en las escuelas. En primer lugar, observamos que, con un curso relativamente corto de didáctica de la programación, la mayoría de los docentes se animó a poner en práctica experiencias acotadas de enseñanza. En segundo lugar, advertimos que la propuesta de desarrollar videojuegos activó dinámicas de clase deseables tales como el interés, la colaboración y la integración de todos y todas las estudiantes. La programación durante muchos años estuvo rotulada “para inteligentes”, “para unos pocos”. Estas experiencias muestran que es posible democratizar este conocimiento en el ámbito escolar y que, además, la lógica de la programación genera un involucramiento con la tarea por parte de la mayoría de los estudiantes. Sosteniendo las premisas de la educación como motor del

desarrollo del pensamiento crítico donde éste implica reflexión, análisis, reconstrucción, desarrollo de modelos, identificación –entre otros, y donde alfabetizar significa utilizar todo el potencial de las herramientas de comunicación esenciales para interactuar en sociedad, es que necesitamos ampliar la mirada sobre la enseñanza de la computación para incluir a la programación.

Esta investigación exploratoria nos abre nuevas preguntas, ¿qué hace que los niños con necesidades especiales tengan un alto rendimiento cuando les proponemos programar? ¿Cuáles son los patrones de trabajo colaborativo que observamos cuando los jóvenes programan en grupos? ¿Cuáles son los usos del tiempo escolar en las clases donde se enseñan programación? ¿Qué conceptos transversales podemos abordar con la programación? ¿Qué proyectos tecnológicos podemos desarrollar con el aporte de computadoras programables? ¿Cómo son los intercambios entre docentes y alumnos cuando enseñamos programación? Estas preguntas, que surgen a partir de observar que la programación genera interés en los alumnos, habilita la integración de disciplinas, permite integrar a estudiantes con capacidades especiales y ofrece una nueva oportunidad para trabajar colaborativamente.

Referencias bibliográficas

- Aloi, F.; Bulgarelli, F.; Palumbo, N. & Spigariol, L. (2016) Corrección automatizada de programas como recurso pedagógico. En XI Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET 2016).
- Casablancas, S.; Caldeiro, G.P. y Odetti, V. (2016) La mirada de los sujetos de educación secundaria en los nuevos escenarios educativos. ¿Qué cambió a partir de la llegada de las netbooks de Conectar Igualdad? III Congreso Internacional de Educación de la Universidad Nacional de La Pampa, Argentina.
- Dutilh Novaes, C. (2011) The different ways in which logic is (said to be) formal. *History and Philosophy of Logic*, 32(4), 303-332.
- Echeveste, M.E. (2017) Situaciones escolares de jóvenes que aprenden programación. Una posición activa del sujeto del aprendizaje. XII Congreso de Tecnología de la Educación y Educación en Tecnología. Universidad Nacional de La Matanza. Buenos Aires.
- Guzdial, M. (2008) Education Paving the way for computational thinking. *Communications of the ACM*, 51(8), pp. 25-27.
- Greco, M.B. (2012) La autoridad nuevamente pensada. En *Emancipación, educación y autoridad. Prácticas de formación y transmisión democrática*. Buenos Aires: Noveduc.
- Grover, S. & Pea, R. (2013) Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), pp. 38-43.
- Lago Martínez, S. (2015) *De tecnologías digitales, educación formal y políticas públicas: Aportes*

- al debate*. Buenos Aires: Ed. Teseo.
- Lee, I.; Martin, F.; Denner, J.; Coulter, B.; Allan, W.; Erickson, J., ... & Werner, L. (2011) Computational thinking for youth in practice. *Acm Inroads*, 2(1), 32-37.
- Levis, D. (2007) Enseñar y aprender con informática/ enseñar y aprender informática. Medios informáticos en la escuela argentina. *Actas de Periodismo y Comunicación*, Vol. 2, N.º 1, diciembre 2016. Disponible en: https://tecnoeducativas.files.wordpress.com/2010/07/u2-y-3-levis_pav06.pdf
- Martínez López, P.; Bonelli, E. y Sawady O`Connor, F. (2014) El nombre verdadero de la programación. Una concepción de enseñanza de la programación para la sociedad de la información. Anales del 10mo Simposio de la Sociedad de la Información (SSI'12), 41ras Jornadas Argentinas de Informática (JAIIO '12), p. 1-23, setiembre 2012. Disponible en: <http://elaulayeltrabajo.proyectoslibres.unq.edu.ar/images/3/35/MartinezLopez-Bonelli-Sawady.pdf>
- Papert, S. (1980) *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books, Inc.
- Papert, S. (1995) *La máquina de los niños: replantearse la educación en la era de los ordenadores*. España: Paidós.
- Resnick, M. (2008) Fallen in Love with Seymour's ideas. Presented at a Special Session of the 2008 American Educational Research Association Annual Meeting.
- Vygotski, L.S. (1999) *Pensamiento y lenguaje: teoría del desarrollo cultural de las funciones psíquicas*. Buenos Aires: Fausto.
- Wing, J.M. (2006) Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Zapata-Ros, M. (2015) Pensamiento computacional: Una nueva alfabetización digital. *Revista de Educación a Distancia* (46). Disponible en: <https://revistas.um.es/red/article/view/240321>